In the claims

1.   (Currently Amended) A system for analyzing a computer application while it is executing without terminating or interrupting the application, comprising:

an application that is executing to be analyzed;

an administration client;

an object shell console executing on the administration client, the object shell console connected to the application so that it can extract information from the application that defines at least the basic internal structure of the application including at least one object component without interrupting the application or causing the application to terminate; and

a graphical user interface presented by the object shell console for presenting at least a portion of the extracted information that defines the internal structure to the user and allowing the user to obtain additional detailed information regarding the internal structure including values associated with the at least one object component.

2.   (Original) The system recited in claim 1, wherein the extracted information includes methods invoked by the application.

3.   (Original) The system recited in claim 1, wherein the extracted information includes variables names and variable values used in the application.

4.   (Original) The system recited in claim 1, wherein the object shell console determines a number of times a selected method is invoked.

5.   (Original) The system recited in claim 1, wherein the object shell console determines a length of time required for a selected method to execute.

6.   (Original) The system recited in claim 1, further comprising a thread that connects the application to the object shell console.

4

7.   (Currently Amended) The system recited in claim 6, wherein the thread is created using ~~Java~~ <u>JAVA programming language</u> RMI.

8.   (Currently Amended) A method for analyzing a computer application while it is executing, comprising the steps of:

      connecting an object shell console to an executing computer application;

      extracting information <u>defining the internal structure of the application and including at least one object component</u> from the computer application without interrupting or terminating the computer application; and

      displaying the information to a user in a graphical user interface.

9.   (Original) The method recited in claim 8, further comprising the step of using a thread to make the connection.

10.   (Original) The method recited in claim 8, wherein the extracting step comprises the step of extracting one or more of a variable name, a variable value, an argument name, an argument value, number of times a selected method is invoked, an execution time of a selected method and a class.

11.   (Original) The method recited in claim 8, further comprising the step of displaying the information in according to a hierarchy.

12.   (Original) The method recited in claim 11, further comprising the steps of:

      displaying at least one class in the graphical user interface; and

      displaying at least one method corresponding to at least one of the at least one classes.

13.   (Currently Amended) The method recited in claim 8, further comprising the steps of:

      accepting a request from the user for more detailed information <u>about the at least one object component including a value associated with the object component</u>;

obtaining the more detailed information <u>including the value associated with the object component</u> in response to the user's request; and

displaying the more detailed information <u>including the value associated with the object component</u> to the user in the graphical user interface.

14.   (Currently Amended) A system for analyzing a computer application while it is executing, comprising:

a computer application executing on an application server;

a thread for connecting an object shell console to the computer application;

means for extracting information ~~related to~~ <u>that defines the internal structure of</u> the computer application <u>including at least one object component</u> without interrupting or terminating the computer application;

means for displaying the extracted information <u>including the at least one object component</u> to a user in a graphical user interface;

means for accepting a request from the user for additional detailed information <u>including a value associated with the object component</u>; and

means for providing the additional detailed information <u>including the value associated with the object component</u> to the user in the graphical user interface.

15.   (Original) The system recited in claim 14, further comprising means for determining an execution time of a selected method.

16.   (Original) The system recited in claim 14, further comprising means for determining a number of times a selected method is executed.

17.   (Original) The system recited in claim 14, further comprising means for extracting one of a variable value and an argument value from the computer application.

18.   (Original) The system recited in claim 14, further comprising means for providing a list of one or more methods comprising the application in the graphical user interface.

19.   (Original) The system recited in claim 18, further comprising means for presenting the information in a hierarchical fashion.

20.   (Currently Amended) A system for analyzing a computer application in real-time, comprising:

an application server on which one or more computer applications is executing, one of the one or more computer applications being a computer application to be analyzed;

an administration client;

an object shell console executing on the administration client that can attach to the application to be analyzed to extract information that defines the internal structure of the application and includes at least one object component from the application to be analyzed; and

a graphical user interface in which the information from the application to be analyzed defines the internal structure including the object component is displayed to a user.

21.   (Original) The system recited in claim 20, wherein the application to be analyzed is written in a computer language that is an interpreter.

22.   (Original) The system recited in claim 20, further comprising a thread through which the object shell is attached to the application to be analyzed.

23.   (Currently Amended) The system recited in claim 22, wherein the thread is created using Java JAVA programming language RMI.

24.   (Original) The system recited in claim 20, wherein the information is one or more of a variable name, a variable value, an argument name, an argument value, an execution time of a selected method and a number of times a selected method is invoked.

Remarks

Claims 1-24 are currently pending and stand rejected. Claims 1, 7, 8, 13, 14, 20, and 23 have been amended. Applicants assert that the claims are now in condition for allowance as set forth more fully below.

Objections to the Specification

The specification is objected to for use of the JAVA trademark incorrectly. The specification has been amended to correct the usage by capitalizing all letters and including the generic terminology. Accordingly, the objection may now be withdrawn.

112 Rejections

Claims 7 and 23 are rejected for use of the JAVA trademark incorrectly. These claims have been amended to correct the usage by capitalizing all letters and including the generic terminology. Accordingly, these rejections may now be withdrawn.

Claims 1, 13, and 14 are rejected for use of the phrase "additional detailed information." These claims have been amended to more specifically define the subject matter of the claim by referring to values associated with the at least one object component of the extracted information that defines the internal structure of the application. Accordingly, these rejections may also now be withdrawn.

102 Rejections

Claims 1, 2, 4-6, 8, 9, and 11-22 stand rejected under 35 USC 102(e) as being anticipated by Ramraj (US Pub 2002/0174174). Applicants respectfully traverse these rejections.

As a representative example of the independent claims 1, 8, 14, and 20, amended claim 1 recites, in part, an object shell console executing on the administration client, the object shell console connected to the application so that it can extract information from the application that defines at least the basic internal structure of the application including at least one object component without interrupting the application or causing the application to terminate. Claim 1 further recites a graphical user interface presented by the object shell console for presenting at least a portion of the extracted information that

8

defines the internal structure to the user and allowing the user to obtain additional detailed information regarding the internal structure including values associated with the at least one object component. Thus, claim 1 is directed to extracting information defining the internal structure of an application, as opposed to other information related to the general execution of an application such as start and stop time, and presenting it within a graphical user interface to the user.

Ramraj discloses monitoring a web based network transaction by utilizing monitoring code linked to the application performing the network transaction and reporting the information obtained from the monitoring to another location. However, the information that is obtained by the monitoring is not information that defines the internal structure of the application including at least one object component. To the contrary, the information obtained by the monitoring in Ramraj is just the information related to a particular transaction, such as the start and stop time of the transaction, the time zone where the transaction occurred, and the operating system of the client machine involved in the transaction. There is no extraction of information defining internal structure of the application itself that includes at least one object component.

Because Ramraj fails to disclose these elements of independent claims 1, 8, 14, and 20, these claims are allowable over Ramraj, singly or in combination with the other cited references, for at least these reasons. Dependent claims 2-7, 9-13, 15-19, and 21-24 depend from allowable base claims and are also allowable for at least the same reasons. Furthermore, one or more of these dependent claims recite additional features allowable over Ramraj, singly or in combination with the other references, such as claim 2 that refers to a method being included in the information defining the internal structure, where the method.

103 Rejections

Claims 3, 10, and 24 stand rejected under 35 USC 103(a) as being unpatentable over Ramraj in view of Lahr (US Pub 2002/0046273). Claims 7 and 23 stand rejected under 35 USC 103(a) as being unpatentable over Ramraj in view of Vaitheeswaran (US Pat 6,687,702). Applicants respectfully traverse these rejections.

As discussed above, these dependent claims depend from allowable base claims and are also allowable for at least the same reasons. Furthermore, the cited references fail to overcome the deficiencies of Ramraj noted in the discussion above.
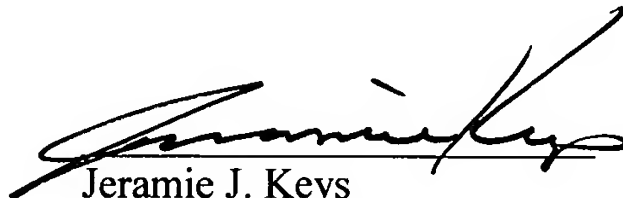
Conclusion

Applicants assert that the application including claims 1-24 is now in condition for allowance. Applicants request reconsideration in view of the amendments and remarks above and further request that a Notice of Allowability be provided. Should the Examiner have any questions, please contact the undersigned.

No fees are believed due. However, please charge any additional fees or credit any overpayment to Deposit Account No. 50-3025.

Respectfully submitted,

Date: January 5, 2005

Jeramie J. Keys
Reg. No. 42,724

Withers & Keys, LLC
P.O. Box 71355
Marietta, Ga 30007-1355
(404) 849.2093

10